Now Including SyncWare News!

Quantum Levels

JUNE 1989

vol. 2, no. 5

TIPS AND TECHNIQUES FOR QL COMPUTERISTS

A New Face for Quantum_Levels

The Timex/Sinclair computing world has reached another crossroad. This issue incorporates Quantum_levels and its sister publication SyncWare News into one magazine. This has been done for financial reasons. Don't worry though, all monies for subscriptions are kept in escrow for each issue and in fact the magazines do pay for themselves. We have seen the trend of expenses going up over the past six years and managed to keep the price of the subscription constant throughout this time period. That is no small task. We have indeed worked on the expense problem and have been able to make ends meet.

Fest Activity

We will have a report on the Washington Computerfest in the next issue and inform all readers as to "How alive" Sinclair computing really is.

New Products

The English market place continues to bring out new software and hardware for the QL. Digital Precision has announced the arrival of "The Solution," a IBM Emulator for the QL. Sharp's has reported brisk sales of the Solution and MS-DOS v 4.Ø1 (for which he has the cheapest price and newest version in the country). Miracle Systems has released their QL Hard Disk system, which is an all inclusive (interface, 3 1/2" hard disk, case and power supply) which should now allow very large applications to be developed. ABC Electronics has worked the bugs out of their MegaRam board. This device replaces the 68ØØ8 CPU with its bigger sister the Quad 68ØØ8 and expands the QL memory range to 4 megabytes. This board adds ram between 1 and 3 megs (above your Trump card or 512K ram card). Unlike MS-DOS, QDOS is written to support the CPU and therefore the added ram is somewhat implemented with very little hassle. I understand that fitting it inside the case is a hassle though. (Imagine a 3 megabyte computer built into a keyboard!)

We have a review on the Solution in this issue and have the hard disk and MegaRam board coming too. We will report on both of them in the next issue. This issue also contains the long awaited (maybe final chapter) on the QL Saga. Part II is technical enough to allow anyone to fix up there machine with a minimum of effort.

EYEING the QL

Tom Bent

This part of the article is written to be non-technical and in fact somewhat historical in nature. I would like to thank Lee Terry, of Comsat Labs, for his help in identifying the true problem discussed herein. He was on a team that wrote the book that was published as the Motorola MECL Handbook. (ECL, Emitter Coupled Logic is what they build Cray computers with.)

I:The Background

In the heyday of Sinclair Research, Sir Clive was convinced by some of his workers to switch from his favorite processor chip, the Z80 CPU, to the MC68000 series microprocessor. Against his better judgement, he agreed.

As work proceeded on the QL, which at that time was called the Pandora (aptly so), Sinclair Research (SRL) came upon some technical difficulties which required surmounting before work could continue.

As SRL saw it, the major difficulty was getting the microdrives to work reliably. This required both hardware and software refinement and redesign of the microdrive tape case.

Timing in the design and production of the QL was just awful. The QL was developed at a time when disk drives were outrageous in price and **sold** at a time when the price of computers **and** peripherals were dropping through **the** floorboards. No one saw this coming. SRL was not alone in the computer glut of '83 and '84. Timex, Commodore, Atari, most CPM computer manufacturers (in this country), BBC and many others took it on the chin too.

As with most computer technology developed in the 80's, the QL with microdrives was obsolete when it hit the market in the summer of 1984. By mid 1985, most of the bugs had been worked out of the QL and its performance was satisfactory.

SRL was still plagued by a high number of returns (as evidenced by the large number of spare boards available and labelled as non-working). Still many more machine had problems with blown out ZX8301's (video display and memory management chip). This is by far the most serious problem for QL owner's everywhere. Because of the short supply of 8301 chips if you lose your video, you might lose your QL.

The QL was originally supposed to have a battery backed up real time clock. SRL couldn't make it work reliably. This turns out to be the same problem as with the 8301 (although not a fatal problem).

Samsung, the makers of the American QL's, noted a few problems early in their production of the American lot. They made attempts to correct the problems, some of which were correct and some of which caused other failures to occur.

Sir Clive in an interview in 1987, said he was sorry that he was swayed to the 68008 CPU for the QL. I personally am not sorry he was. The 68008 has several advantages over Z80 and 8088 chips when it comes to multitasking and memory management. (At least it does if you have to work around Microsoft operating system code).

What SRL did not know (and is poorly documented in early Motorola literature) is that the 68008 chip has special high speed logic for handling the data bus. The 68008 usually must chunk in data 4 times before it can act on it, because it has a 32 bit internal structure. (The QL has an 8 bit data bus.) To keep performance up, Motorola felt high internal speed was necessary.

It is not that this high speed is bad, mind you, but rather the speed with which both the data and address bus' change logic state from 5 volts to 0 volts. The problem is that the logic doesn't stop when it gets to 0 volts! It continues down to -1, -2 or even -3 volts, before it recovers to 0 volts. This phenomenon is now called "Ground Bounce."

This transition is very fast. The QL has a fall time of 3 nanoseconds. Very recently Bob Curnutt, of Harry Diamond Labs, using a 1 gigahertz scope measured this fall time at 1 nanosecond. [I would be tempted to call this a voltage spike.-ed] A fully loaded QL has a rise time (the time for the logic to go from about 0.5 volts to

4.5 volts) of about 30 to 40 nanoseconds!

By comparison, the 8049 co-processor (similar to the Z-80 in logic) has a rise time of 90 nanoseconds. This processor runs at 11 megahertz (MHz) versus the 68008 which is rated at 8 MHz. The Z-80 is typically rated a 4 MHz and has rise times much slower, but the TS2068 runs at 3.57 MHz and the ZX-81 runs at 3.25 MHz (an Apple II runs at 1 MHz). The PC clones run from 4.7 up to 10 MHz with fast enough RAM chips installed.

OK, we have labeled the QL's inherent problem, now what causes it and can we fix it? The answer to question 2 is yes (see II: Foreground). The problem is caused by very high instantaneous currents going through thin PC board traces (especially ground traces).

The 68008 CPU consumes 280 milliamps, which sounds reasonable by CPU standards, but most of that power is consumed in the scant nanoseconds of each transition. The ground bounce comes from a summation of currents of each individual bus line being returned on the ground simultaneously.

An analogy is lighting in your home. You can cut the wire going to a light and install a switch anywhere in the circuit. When the switch is open (even installed on the common ground) the circuit is open and the light won't light up. You can think of the QL as about 200 little lights with individual wiring on the power side (bus lines) and all connected together with 2 main ground lines on the other side. I don't want to get too technical, but you can get the drift.

Overcoming this problem is relatively easy (once you figure it all out). I'm surprised that SRL had such a tough time of it.

This portion of the article is somewhat more technical and in fact hands on in a manner of speaking. In the vein of past Q_TIPs articles, this one is quite intense, but it does end a two year saga of learning what makes the QL tick and what makes it stop ticking. I used some good equipment to tackle the problem, mainly a Hewlett Packard 1722B scope (280 MHz bandwidth) and a Metex M-4650 4 1/2 digit DMM (available from Jameco Electronics for $99.00 and a much recommended addition to your workbench).

I must say up front that not every QL has every problem, nor is every QL "bad" and likely to fry your peripherals. However, every QL does have the ground bounce problem to some extent (guaranteed).

II: Foreground

This is a tough section to write, because I want to explain why the problem exists as we actually go through the how-to-fix motions. Let's tackle one step at a time.

Problem:

There is a large variance in the ground levels between the 68008 and the 8301. These two chips are the big ones, side by side on the left side of the QL PC board. Although these chips are a scant 0.2 inches apart, this ground mismatch can be as high as 3 or more volts. Samsung decided to cut the ground connection between the two chips. This is evidenced by a trace cut in the front left corner of the PC board near DC17. This did stop a ground loop for the data bus (a mismatch in trace length between the 68008 and IC21, the 74LS245), but it created a ground loop in the very short control lines directly between the two chips.

Solution:

Some QL's have this trace cut, others have later PC boards with the trace removed from the mask and early boards (about 800 machines) don't have this trace cut. If it isn't cut, cut it. As mentioned in Q_L issue 2:4, measure the voltage at the CPU (the cap on top) and then measure it at the regulator (the two left pins at the chip on the heat sink. If the difference greater than

0.08 volts (revised from 2:4), the get the following three parts:

1 10 microfarad tantalum cap (10 volts or greater)
1 .01 microfarad ceramic cap
2 .001 microfarad ceramic caps

Attach the 10 mF tantalum cap to the top of the CPU (plus side to pin 13 and minus side to pins 15 and 35) in parallel with the .022 mF cap already there. Careful, reversing the Polarized Tantalum Cap will fry your 8301 when the cap eventually fails. (We found out the hard way.) This cap keeps 5 volt supply to the CPU virtually flat. This value can supply the instantaneous current that the 68008 desperately needs.

Attach the .01 mF cap from pin 6 to pin 15 on the ZX8301 (IC22). This is the proper bypass value for this chip. DC18 is simply too far away to effectively decouple to 15 MHz bopping around inside this chip. If your video breaks up and goes all white on you, this simple 3 cent capacitor would have prevented a $30 chip replacement.

Attach one .001 mF cap from pin 6, IC22 to pin 35, IC18. This in effect couples the two ground traces without connecting them together. If ground variance gets too large, then current will flow through the cap, keeping them level (i.e., less than 1 volt p-p). This also changes time constant of the ground loop from 10 nanoseconds to 40 nanoseconds! This causes a drastic improvement in the reliability of the QL.

OK, so what's the other .001 cap for? It's for the real problem QL's, but it won't hurt the really good ones. Make sure it is placed EXACTLY where I say. There is a feed through 0.5 inches to the left of IC33 pin 14. That is the first ROM chip. It's the thick ground trace that ends just below the 2 in C2. Attach the other .001 cap from this feed through to pin 6 of IC22. This does the same thing for the other ground loop that wasn't previously addressed by anyone.

Don't get smart and directly connect the grounds of chips together. You will fry everything.

Problem:

My QL crashes every 5 minutes (or every 30 minutes or 45 minutes).

Solution:

Follow the previous steps and don't forget what we wrote in 2:4. Add that 20 Ohm resistor from the peripheral connector pin 32 to the 5 volt bus (the jumper wire on the left side). Resolder the 3 pin power connector and resolder or replace all 4 wires on the spider board (above IC 1, 2 and 3). If you are in doubt of the integrity of your 5 volt regulator, just replace it. It must be 4.950 volts or greater.

Attach a 20 or 22 gauge jumper wire from the left pin of the voltage regulator to pin 8 of IC 5 or IC 6. This will provide a star power distribution point rather than a branch distribution of the 5 volt bus.

Problem:

My QL goes through an 8301 chip every 3 months (or every 6 months or every year)

Solution:

Pick up 8, 270 picofarad capacitors (range 200 to 330). If it is one 8301 every year then 3 or 4 caps will do. Begin soldering each one on a ram chip. Start at IC8, from pin 14 to pin 8 (data bus line to 5 volts). Then go to IC15 (skip IC16) and repeat. The IC14 and so on. IC15 is the most critical (that's data line D6). D6 has the most screwed up path I ever followed. Sinclair should have shot the guy who laid out that path. The scope trace on this line has more ghosts than a good dungeons game.

Problem:

I get a lot of jibberish at 1200 baud on my modem and the QL has trouble receiving at 9600 baud.

Solution:

This is the last frontier. The 1489 chip, IC26 the the serial port receive chip. Sinclair saw fit to leave the logic 0 transition bias pins unterminated. This means that when you receive a less than optimum signal, it may cause a false trigger of a bit, and just print gibberish on the screen. Connect a .01mF cap from pin 5 IC26 to pin 7 IC26 and from pin 12 to pin 14 IC26. This will stabilize the ser2 input. You can repeat this on pins 2 and 9 for the ser1 inputs. If you are dropping characters, it's another story. This problem has to do with the keyboard queue. By the time the keyboard is read, the interrupt is gone or otherwise lost and the entire character is lost. I have to leave this to a good software hound (how about it Tony?).

If you put a scope on the power line near the ROMs you might see a large spike on the power bus twelve times a second or so. This phenomenon is caused by having the 8049 crystal and the clock CPU (7.5 UK, 7.5545 US) be in sync and have all of the data and address lines on both CPU's firing at the same time. Sinclair did not put a bypass cap on this CPU (simply ridiculous). Add a .1 mF cap between pin 40 (5 volts) and pin 20 (ground). Add an optional 10 Mf tanalum cap in parallel with the .1 cap for perfection, but watch the polarity.

Add a .01 mF. cap between pin 11 and pin 20 of the ZX8302, IC23. Although the pads are present for this chips' bypass cap, Sinclair saved a fraction of a penny here too.

Problem:

My clock board works like SH_t!!

Solution:

Hmm, this is a tough one too. The Qimi board works very well, but goes dead in a week. The TK board works well in British machines for the most part, but not always. Most other fixes don't help anything.

First, it is not the ZX8302. It is the QL mother board. When the 68008 fires up, the data bus and ground plane experience ground bounce and this negative voltage excursion avalanches the CMOS circuitry and zips it all to 0. Adding capacitance to this area directly affects the performance (and temperature) of the CPU. Some machines can tolerate 100pF on each data line, others cannot. I'm going to try 47pF on each data line as a generic fix, but it depends on each machine. The data path is just too long. The best fix should include a 74LS245 buffer on this chip, isolating pins 12, 22,23,24,36,37,38 and 39 from the outside world. I would go even further and say a National Semiconductor LS245, because they are "slow" chips.

Hardware designers note: Do not build add on devices with FAST logic. These devices just create too much extra ground noise. If you do, then please add bypass caps to your boards. I recently saw a PC keyboard interface that had a "problem" in this vein.

oooooooooooooooooooo

Call For Articles

We need your articles, reviews, reports on applications, etc. Let's keep ourselves informed, but you too can help. Send your articles (anywhere from a paragraph to several pages) on magnetic media. Basil and I have not yet worked out the logistics for 1ØØØ, 2Ø68 and QL, but it should remain as QL information goes to Columbia, Maryland and 1ØØØ and 2Ø68 information goes to Bloomington, Indiana.

THE SOLUTION:

An IBM Emulator for the QL

James Hunkins

The variety of software for the QL, while not nearly as extensive as for IBM family machines, fits the needs of most QL owners, at a much lower price than their IBM counterparts.

Occasionally, however, QL owners (as with other computer users) have need to run software that is not available for their machine. Examples of this might be specialized analysis programs or, as in my case, certain utilities from work. With the exception of ASCII document processing made available from IBM disk reading programs and file transfers through modems or laptops, there has been no way short of buying another computer.

A new, less expensive alternative has now been introduced by Digital Precision called, The Solution.

EMULATOR DESCRIPTION & LIMITATIONS

The Solution, an IBM emulator, by translating code and supplying its own version of the IBM Bios (machine level operating system software) can make the QL computer think that it is an IBM clone and run most IBM compatible software.

An important limitation of all emulators should be noted at this point before you rush out and slap down your $150 for Solution (the evaluation copy was purchased from Sharp's) plus the additional $70+ for MS-DOS (an IBM (Disk) Operating System).

Emulators have to actually read the code which is being run and then translate it to the native code of the computer it is being run on. As IBM clones use Intel 8x86 family micro-processors and the QL uses the Motorola 68008, not only does the emulator have to look up conversions as it runs the code, but in many cases it has to replace the original instruction with several Motorola instructions. Additionally, conversions between operating system drivers have to be performed, such as screen and keyboard handlers.

Obviously, even with the 7.5545 MHz clock speed of the QL, an emulator is going to be much slower than the machine it is trying to be (faster QL clones help the situation somewhat).

With a speed of 10 to 20% of the IBM 4.77 MHz XT models (as per Digital's claim), you can immediately rule out using this emulator with interactive games or any program that has many layers (such as shells and interactive development environments.)

On the other hand, many programs, especially those that require human intervention, such as word processors and spread sheets, may run fast enough to serve your needs. Even some programs that take a while, such as some assemblers and compilers, can be run while you do something else.

Obviously, the need for an IBM emulator varies tremendously and you have to determine if the programs that you need will run at a sufficient speed. Also, be aware that some programs, expecially cheap and some public domain software, tend to do things illegally by bypassing the normal operating system. This will often cause an emulator (along with many IBM clones) to crash.

Again, it is a case of buyer beware as no company can possibly guarantee their emulator (or computer for that matter) to run all software.

Before this scares you away please note that that most popular and/or expensive IBM programs are completely legal and will run on a good emulator.

IBM COMPATIBILITY

While on the subject of 'good,' just how well does Digital Precision's Solution program perform? (You will need at least one disk drive, 5 1/4 or 3 1/2, and 256K or more expansion memory to run Solution.)

As to IBM compatibility, it seems to find itself in the league of emulators that have been on the market for other machines for years. Of the over 70 programs Digital tested, five in the list they supplied failed. Three of these were games which illegally read the keyboard, probably to gain speed. Two surprise failures were BASIC and BASICA, both Basic Interpreters from IBM. The problem here is that each of them uses copywrite BIOS routines that are only found on non-clone IBM's. Digital solves this by simply supplying the public domain PDBasic instead with Solution at no extra charge.

A pleasant surprise was that when 'tough' programs were tested with Solution, most ran, including MathCad. (The menu box on the screen top was difficult to read.) Tutsim (system engineers take note) worked, along with DAC Easy Write Word Processor (too slow to use), the Borland Turbo C Graphics Demo, Quick Dos Shell and Editor, PCLink (connection to the Z88 portable), Shogun (too slow to play but nice graphics), and the PC Magazine Lab Benchmarks.

The few failures included Top Gun, Express Graph, and 2D Plot. In each case, the program either locked up the keyboard (illegal type code) or Solution came back saying that it did not recognize a piece of code.

This message usually results from code written for 80286 or 80386 machines, or self modifying code (a touchy subject in itself).

The PC Magazine Lab Benchmarks gave a performance rating of 8 to 12% that of a 4.77 MHz XT computer. As most programs also use the BIOS which in this case is handwritten in native 68000 assembly, eliminating translation need, the 10 to 20% figure given by Digital is a realistic figure.

As was discovered during testing, even though you can literally boot up Solution directly on the QL (they even generate the Boot file automatically when you make up a back up copy), reading the manual is a MUST. There are some (options) that you may want or even be required to change, to run some programs. These include how you want to use your disk drives and the option of interrupts and tracing being on. Not reading the manual first will make life a lot worse than it really is.

SCREEN HANDLING, DISK DRIVES

One of the most difficult parts of an emulator is the screen control. Solution has a full and accurate CGA emulation, including CGA register compatibilty (many programs 'cheat' and use the registers directly to speed up the screen) through a unique scanning concept. It runs the screen control as a separate job, set at a low priority. Periodically scanning the screen for needed updates, it automatically adjusts its own job priority upward as necessary to handle the work load.

A few minor irritations occur from this method however. During an update, you sometimes see the scan take place, often seemingly out of normal sequence. Additionally, keystrokes will have a short delay before they are shown on the screen.

By getting used to the type ahead buffer (the same as the QL uses), you can easily handle this and actually somewhat improve the DOS performance. Care must be taken to not type entries in more than once, as it might confuse some programs.

The CGA emulation is also exceptional in a few other aspects. The screen handler also checks for two like colors existing side by side (the QL does not allow for the intensity shading that CGA has) and will change one of them to avoid blank areas on the screen. It also uses the QL stipple ability to provide different shades of colors. (The Borland Turbo graphics demos and some game screens are good places to see this.)

Along with text, graphics mode is fully supported. However a smaller screen area is used for this. The resolution is surprisingly sharp.

As to disk drive usage, if you want to keep a drive separate from DOS for QL useage, you can do so. As with DOS drives, you have a choice of 40 track (compatible with IBM) or higher density 80 track (depending on the type of drive in use).

SOLUTION EXTRAS

Solution offers a method of, either directly from DOS or from within many DOS programs, actually stopping everything and changing the configuration and options of the system. This does not exist in DOS.

You can modify the disk drive configuration and even control timing interrupts and trace (normally disabled to speed up the system). Options are given to break out of a program back to DOS or to either reboot DOS or to return to the QL operating system. An unusual option is to ask Solution to tell you when it finds an unknown code, at which time it gives the choice of continuing or aborting the program.

It is suggested that while software is being checked for compatibility that interrupts and trace are kept enabled. Some software will not run otherwise. This will also allow you to get out of a program gone bad without having to reboot the entire system.

Solution allows the user to multitask itself with QDOS programs or even another copy of Solution (another DOS program). The QL normal method of switching from job to job is used but with a different key stroke to avoid conflict with a standard DOS key sequence. The same screen/window conflicts occur here as to in normal QL mode.

Solution, in addition to supplying a public domain version of PDBasic, includes XOVER, a series of QL/DOS disk handling utilites, allowing the QL (in normal mode) to access both QL and DOS format disks. It has features similar to Discover.

DOCUMENTATION

The documentation that comes with Solution is complete but should be rearranged. Specifically, the items discussed at the end of the manual need to be known before many programs will run. I could not even load the DOS disk until I read the last few pages. Of course, the entire manual should always be read before trying any new program (which we all do so faithfully!)

The contents are comprehensive and much explanation is given to just how the emulator works. This is useful if you happen to be stuck with some 'bad' programs that will not run. Often, once you understand where they misbehave, you can find ways around the problem and eventually get them up and running.

DIFFICULTIES

Being the first version of the program sold, some problems are bound to creep in. None of those found were unsolvable, and once they were known they were easily avoidable. Knowing Digital Precision's reputation, the problems will be fixed in a future release.

CONTINUED ON PAGE 17...

The SyncWare Pages

Basil Wentworth, Editor

CUSTOM TAPE/DISK DIRECTORY

Bob Swoger
Motorola, Inc.
Schaumburg, IL
312/576-8068

T/S 2068, T/S 2040, T/S 2050,Legend 1080, TASMAN CPI, LARKEN DOS

Work Smarter - Not Harder

This article describes a way for owners of the LARKEN disk system to get around from program to program more easily in a multi-drive system. Larry Kenny has made a disk drive system possible for many T/S 2068 owners by making an interface for about $120 while drives can now be found used for under $50. This makes his disk drive system for under $200 comparable to the going price of a floppy drive system used on other 8 BIT color computer systems. If you never had the ability to run Spectrum programs because you didn't want to open the case of your precious T/S 2068, Larry's system fixes this problem by providing a place on his controller for a Spectrum ROM and a Kempstone Joy-Stick on his floppy drive interface. Adding his 250K RAM-DISK to my system has really brought me a long way from my old slow tape system.

The program presented here provides a way to find programs filed away on disk or tape and manage their movement within or between storage cases with relative ease.

Even without a disk drive system, this article is very useful for those still using a tape system on their machines or those who wish to file audio or video disks or cassettes. Because this program was first written as a tape directory the program presented here is the cassette tape version with modifications given to change it to the LarKen disk directory version.

After years of collecting all sorts of program tapes for my T/S 2068 I began to have trouble finding the program I wished to run amoung the many boxes of tape programs I had aquired, and the list of new programs is still growing! This program has solved my problem by giving me a way to move titles in boxes from slot to slot with great ease. I then store this program in the front slot of the first box and run it first to find other tapes. I also use it to manage my VCR tape drawers and data disk library cases. I store the disk version on my LARKEN RAM-DISK for quick access. I used an earlier version of this program with my ZX81 as the first program on a tape of many programs to show the programs on a single tape and to load the desired program from it. Some may wish to modify this program to do these things.

Each displayed board represents a storage box containing 20 slots. The 20 slots are divided into two columns of ten slots each. Please number these in your mind as 1 thru 10 on the left side and 11 thru 20 on the right. You may put titles as long as 12 characters into each space provided. You may then Add new titles, Delete titles, or Move a title from one slot to another slot either in the same box or a different box. The ability to see the titles in each box is as easy as hitting the 'NEXT' key or using the LEFT or RIGHT ARROW keys. A SAVE feature with VERIFY is also provided. LPRINT key will print the contents of all 12 boxes to your T/S 2040 printer and the COPY key will send a picture of the box on the screen to the printer. This can be taped to the inside of the clear plastic top of the tape storage box to show what is inside.

SCREEN TRICKS

Now for a few tricks. In her book "T/S 2068 BASICS and BEYOND", Sharon Zardetto Aker states that if we try to print to the 23rd and 24th line of the screen the system will crash. Line 820 shows one of three ways to do this without crashing. The second is an input statement in the form of line 1020 and the third is with the use of the statement PRINT#1,"string".

(Editor's NOTE: anything PRINTed in lines 23 and 24 will be erased by any report message received. Thus, if you command PRINT#1,"Basil loves Jocelyn", the only output you will see will be

0 OK, 0:1

-- or whatever line number you actually used -- before you have a chance to see the mesage you asked for. The way to avoid this problem, of course, is to insert some command that will delay the report message, such as a PAUSE, or a statement like

```
100 IF INKEY$="" THEN GO TO 100
```

With the technique of Line 1020, the delay is automatically provided by the fact that the computer is waiting for an INPUT.)

Custom Keyboard Section

The CUSTOM KEYBOARD SECTION sets up custom features for your machine even after exiting the MENU program as long as the machine is not turned off or another program doesn't reset the feature. In this section you may enter your favorite POKEs to adjust key 'click' or 'repeat' key hold down duration.

My favorite feature of the CUSTOM KEYBOARD SECTION took me a little time to find but was well worth the effort. Since I have been writing BASIC programs for about 16 years now, I was accustomed to having my terminal come up in the CAP or upper case mode to answer prompts. This is accomplished in line 8040 with the POKE 23658,24 statement. POKEing this flag to 16 will cause the machine to return to the lower case mode.

A 'c/o' key is provided on the GRAPHICS 'B' key in the CUSTOM KEYBOARD SECTION as are a few other things. The GRAPHICS 'A' is set-up to provide the interesting background for the MENU board. This key is provided on the ZX81 but not the T/S 2068 and is a nice background on monochrome monitors. It will look purple on a color monitor although it is truly black and white!

Now for a few words about abbreviating. As I stated earlier, there is room for only 12 character titles. To fit in longer titles, try leaving out first the vowels, then one of the consecutive double letters in a word. If the word starts with a vowel, leave it in. Most long titles will still make sense. You can use this trick for many other data records in filing systems. If someone misspells a word, it is often because the wrong vowel has been used. The ancients could read and write without vowels (Hebrew and Arabic are still customarily written without vowels--ed.) and you will find that, knowing the rules, you will too. Also, it does save BYTES!

Help Screen

A HELP screen is also available to show all the possible keys that will give you desired results. Just press 'H' to see the HELP file.

I suggest that when storing tapes or disks you try to keep a particular theme in each box, utilities in one box, games in another, word processors in another, terminal programs in another, etc. In this way locating a program is made easier. My recently aquired disk library-cases hold 10 disks each. I backup my LARKEN RAM-DISK on disk number 0 and then put disks 0 through 9 in the first case, 10 through 19 in the next case and so on. I then use slots 11 through 20 of the menu board to comment or expand the description for the first 10 slots.

To change the program for use as a LARKEN disk directory change the following lines:

```
 610 REM *CUSTOM DISK DIRECTORY*

9993 PRINT #4: SAVE "DISKS.B1" L
INE 8000

9995 POKE 23659,0: PRINT AT 22,0
; PAPER 4;'"Now preforming CRC "
:POKE 23659,2:PRINT AT 0,0

9996 PRINT #4: VERIFY ""

9998 PRINT #4: LOAD "L.B1"
```

To initialize and save the program, after loading it into the T/S 2068 and making ready your storage device, type the following in the immediate mode:

```
DIM G$(3121):GOTO 9990 <ENTER>
```

Never RUN this program as all the DATA will be erased! Use GOTO 1 instead.

Download This Program

This program can be downloaded from the Motorola BBS using MTERM II on your Westridge 2050. Use the data parameters 8-N-1 Full-DUPLEX, buffer conversion HEX and the buffer closed and empty. Call 312/576-7072 and after connecting hit <ENTER> until you get the '>' prompt. Then type: LOG SINCLAIR/GUEST

Go to the BBS download section and download either DISKS.B1 or TAPES.B1 . Remember that you are a guest on this board and please act accordingly.

```
 61Ø REM *CUSTOM TAPE DIRECTORY*
WRITTEN BY ROBERT E. SWOGER JULY
 23, 1988*
 62Ø PAPER VAL "7": GO TO VAL "8
ØØØ"
 71Ø REM **** DISPLAY Section **
 72Ø PRINT AT A,L; PAPER VAL "7"
; FLASH D;"BOX  ";AT A,VAL "17"+
(I<B);I:  FOR J=D TO B: PRINT  P
APER A; INK VAL "9";AT J*E,D;G$(
```

```
((I*260)-259)+((J*L)-L) TO ((I*2
60)-259)+(J*L));AT J*E,17;G$(((I
*260)-259)+(((J+B)*L)-L) TO ((I*
260)-259)+((J+B)*L)): NEXT J: RE
TURN
 810 REM **** PROMPT Section ***
 820 POKE VAL "23659",A: PRINT A
T VAL "22",A; INK A; PAPER VAL "
4";'" Add  Delete  Move  Next  S
ave": POKE VAL "23659",E: PRINT
AT A,A: RETURN
 920 POKE 23659,A: PRINT AT 22,A
; INK A; PAPER VAL "4";'"  PRESS
  <ENTER>   TO  CONTINUE ": POKE
 23659,E: RETURN
1010 REM **** ADD Section ******
1020 INPUT " ADD MODE - SLOT(1 -
 20)?";J: GO TO VAL "1020+(A<J A
ND J<=C)"
1030 INPUT " TITLE?";S$
1040 IF S$(D)=" " THEN  GO TO 10
30
1050 LET G$(((I*260)-258)+((J*L)
-L) TO ((I*260)-259)+(J*L))=S$:
GO TO VAL "3130"
2010 REM **** ERASE Section ****
2020 INPUT " DELETE MODE - SLOT(
1 - 20)?";J: GO TO VAL "2020+(A<
J AND J<=C)"
2030 LET G$(((I*260)-258)+((J*L)
-L) TO ((I*260)-259)+(J*L))="":
GO TO VAL "3130"
3010 REM **** MOVE Section *****
3020 INPUT "  MOVE FROM SLOT?(1
 - 20)";J: GO TO VAL "3020+(A<J A
ND J<=C)"
3030 LET S$=G$(((I*260)-258)+((J
*L)-L) TO ((I*260)-259)+(J*L))
3040 POKE 23659,A: PRINT AT 22,A
; INK A; PAPER VAL "4";'" DELETE
 FROM OLD SLOT?": POKE 23659,E
3050 LET Y$=INKEY$: IF Y$="" THE
N  GO TO VAL "3050"
3060 BEEP .03,34.4: IF Y$(1)="Y"
 THEN  LET G$(((I*260)-258)+((J*
L)-L) TO ((I*260)-259)+(J*L))=""
: GO SUB VAL "700"
3070 IF Y$(1)<>"N" AND Y$(1)<>"Y
" THEN  GO TO VAL "3040"
3080 INPUT "  MOVE  TO WHAT BOX?
(1 - 12)";I: GO TO VAL "3080+(A<
I AND I<L)"
3090 GO SUB VAL "700"
3110 INPUT "  MOVE INTO SLOT?(1
 - 20)";J: GO TO VAL "3110+(A<J A
ND J<=C)"
3120 LET G$(((I*260)-258)+((J*L)
-L) TO ((I*260)-259)+(J*L))=S$
3130 GO TO VAL "9090"
4010 REM **** LPRINT Section ***
4020 FOR I=D TO L-D
4040 LPRINT ,AT A,L;"BOX  ";AT A
,17+(I<B);I
4050 FOR J=D TO B
4060 LPRINT ,AT J*E,D;G$(((I*260
)-259)+((J*L)-L) TO ((I*260)-259
)+(J*L));AT J*E,17;G$(((I*260)-2
59)+(((J+B)*L)-L) TO ((I*260)-25
9)+((J+B)*L))
4070 NEXT J
4080 NEXT I
4090 GO TO VAL "9070"
5010 REM **** HELP Section *****
5020 PAPER VAL "4": PRINT AT A,A
;"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
  Each BOARD represents a File B
oxcontaining 20 SLOTS, 10 down t
heleft side and 10 down the righ
t,numbered 1 thru 20. Add, Delet
e,or Move NAMES in the slots usi
ngthe Select Keys shown below.

  Option        Select Keys
  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
  ADD             A   +
  DELETE          D 7 -
   MOVE            M 6
    NEXT            N 8 F
    BACK            B 5
    HELP            H
     PRINT           C K L V
      COPY            P Z
       SAVE            S
       END or QUIT     E Q
       "
5030 GO SUB 900
5040 LET A$=INKEY$: IF A$="" THE
N  GO TO 5040
5050 BEEP .03,34.4: GO TO 9000
8010 REM **********************
*** Custom Keyboard Section ****
***
8020 DIM S$(12): REM  DIM G$(312
1)
8030 LET A=NOT PI: LET B=VAL "10
": LET C=VAL "20": LET D=SGN PI:
 LET E=VAL "2": LET L=VAL "13":
REM 0-ROWS-SLOTS-1-2-LENGTH
8040 POKE VAL "23562",E: POKE VA
L "23609",VAL "50": POKE VAL "23
658",VAL "24"
8110 FOR H=A TO VAL "7" STEP E
8120 POKE USR "a"+H,BIN 10101010
8130 POKE USR "a"+(H+D),BIN 0101
0101
8140 NEXT H
8210 POKE USR "b"+0,BIN 01000000
8220 POKE USR "b"+1,BIN 10100010
8230 POKE USR "b"+2,BIN 10000100
8240 POKE USR "b"+3,BIN 10101000
8250 POKE USR "b"+4,BIN 01010000
8260 POKE USR "b"+5,BIN 00101100
8270 POKE USR "b"+6,BIN 01010010
8280 POKE USR "b"+7,BIN 10001100
9010 REM **** BILLBOARD Section
*********
9020 INK A: PAPER VAL "4": BORDE
R VAL "4": CLS
9030 FOR I=A TO VAL "21"
9040 PRINT  PAPER VAL "7";"*U*U*
U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*
U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*
U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*
U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*
U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*
U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*
U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*
U*U*U*U*U*U*U*U*U*U*U*U*U*U*U*U"
9050 NEXT I
9060 PLOT A,A: DRAW A,175: DRAW
255,A: DRAW A,-175: DRAW -254,A:
 DRAW A,174: DRAW 253,A: DRAW A,
```

```
-173: DRAW -252,A: DRAW A,174
9070 LET I=A
9080 LET I=I+D: IF I>=L THEN  LE
T I=D
9090 GO SUB VAL "700": GO SUB VA
L "800"
9110 REM **** INPUT  Section ***
9130 LET I$=INKEY$: IF I$="" THE
N  GO TO VAL "9130"
9140 BEEP .03,34.4
9150 IF I$="A" OR I$="+" THEN  G
O TO VAL "1000"
9160 IF I$="D" OR I$="7" OR I$="
0" OR I$="-" THEN  GO TO VAL "20
00"
9170 IF I$="M" OR I$="6" THEN  G
O TO VAL "3000"
9180 IF I$="P" OR I$="Z" THEN  C
OPY
9190 IF I$="C" OR I$="K" OR I$="
L" OR I$="V" THEN  GO TO VAL "40
00"
9210 IF I$="S" THEN  GO TO VAL "
9990"
9220 IF I$="B" OR I$="5" THEN  G
O TO VAL "9300"
9230 IF I$="H" THEN  GO TO VAL "
5000"
9240 IF I$="E" OR I$="Q" THEN  G
O TO VAL "9997"
9250 IF I$<>"F" AND I$<>"N" AND
I$<>"8" THEN  GO TO VAL "9130"
9260 GO TO VAL "9080"
9320 LET I=I-D: IF I<D THEN  LET
 I=L-D
9330 GO TO VAL "9090"
9991 REM  SAVE Section
9992 BEEP .15,5
9993 SAVE "TAPES" LINE 8000
9994 BEEP .042,50: PAUSE 2.8: BE
EP .042,50
9995 POKE 23659,A: PRINT AT VAL
"22",A; PAPER 4;'"  Rewind & Pla
y tape to  VERIFY ": POKE 23659,
E: PRINT ÁT A,A;
9996 VERIFY "TAPES"
9997 BEEP .042,50: PAUSE 2.8: BE
EP .042,50
9998 GO TO VAL "9000"
```

CUSTOM TAPE DIRECTORY

FRACTIONS

Carl Downer
ZX81/TS1000 16K; TS2068

You remember Alex Sweitzer (Geometry Plotter, SWN Vol. 5, No. 3), who teaches an elementary-school class that works extensively on TS computers. This program was written by one of his fifth-grade students. The author describes his program as follows:

The program will multiply two mixed numbers. First it will change the mixed numerals to improper fractions. Second it will multiply the two improper fractions. Next it will either change the improper fraction to a mixed numeral if needed, or it will put the fraction in lowest terms as many times as needed. When the fraction is in lowest terms it will print the problem and the fraction. Also at the beginning of the program it will give the directions.

EDITOR'S COMMENT

The program is straightforward, and shows a good knowledge of the mechanics of working with fractions. I am tempted to recommend to all arithmetic teachers out there that they grade their students not only on their consistent ability to get the right answer, but also on their ability to write a computer program to arrive at that answer. (Sounds a little like the old Tom Lehrer record on New Math, doesn't it?)

The section on reducing to lowest terms is especially thoughtful.

The illustrations show the screen display at various stages during the computation. The figure with the caption "COMPUTING LOWEST TERMS" is a bit misleading: the numerator and denominator are never divided by 17 as the caption says. Rather, the computer makes a trial division of all numbers (including, in this case, 17) down to two; the division is actually performed only if the trial is successful.

The fact that Carl was able to write this program shows that there is at least one schoolroom in this country where computers are being used intelligently as a teaching tool. I hope that Carl and his classmates will have a chance to continue their studies at such a high level after they leave Alex's class.

$$\frac{25}{4} = \frac{\quad}{\quad}$$

```
COMPUTING LOWEST TERMS...WAIT.

I AM DIVIDING NUMERATOR AND THE
DENOMINATOR BY: 17
```

$$2\frac{1}{2} \times 2\frac{1}{2} = 6\frac{1}{4}$$

$$\frac{25}{4} = 6\frac{1}{4}$$

```
  20 REM CARL E. DOWNER
     3 RD STREET
     NEWELL, PA 15466
  40 SLOW
  60 CLS
  80 REM TITLE PAGE
 100 PRINT AT 10,0;
 120 FOR A=1 TO 12
 140 PRINT "
                "
 160 NEXT A
 180 PRINT AT 11,5;"MULTIPLYING
FRACTIONS"
 200 PRINT AT 16,6;"BY CARL E. D
OWNER"
 220 PRINT AT 20,11;"MAY 1988"
 240 PAUSE 250
 260 CLS
 280 REM SET SPEED
 300 PRINT AT 8,0;"PRESS <F> FOR
 FAST SPEED..."
 320 PRINT AT 10,0;"PRESS <S> FO
R SLOW SPEED..."
 340 INPUT B$
 360 IF B$="F" THEN LET P=60
 380 IF B$="F" THEN FAST
 400 IF B$<>"F" THEN LET P=6000
 420 CLS
 440 GOSUB 3500
 460 CLS
 480 GOSUB 3840
 500 REM GET NUMBERS FOR FRACTIO
N
 520 PRINT AT 0,0;"FIRST WHOLE N
UMBER:";
 540 INPUT FIRST WN
 550 IF FIRST WN<1 OR FIRST WN>9
9 THEN GOTO 540
 580 PRINT FIRST WN
 600 PRINT
 620 PRINT "FIRST NUMERATOR:";
 640 INPUT FIRST N
 660 IF FIRST N<1 OR FIRST N>99
THEN GOTO 640
 680 PRINT FIRST N
 700 PRINT "FIRST DENOMINATOR:";
 720 INPUT FIRST D
 740 IF FIRST D<1 OR FIRST D>99
THEN GOTO 720
 760 PRINT FIRST D
 780 PRINT
 800 PRINT
 820 PRINT "SECOND WHOLE NUMBER:
";
 840 INPUT SECOND WN
 860 IF SECOND WN<1 OR SECOND WN
>99 THEN GOTO 840
 880 PRINT SECOND WN
 900 PRINT
 920 PRINT "SECOND NUMERATOR:";
 940 INPUT SECOND N
 960 IF SECOND N<1 OR SECOND N>9
9 THEN GOTO 940
 980 PRINT SECOND N
1000 PRINT "SECOND DENOMINATOR:"
;
1020 INPUT SECOND D
1040 IF SECOND D<1 OR SECOND D>9
9 THEN GOTO 1020
1060 PRINT SECOND D
1080 PAUSE 300
1100 CLS
1120 REM PRINT TO SCREEN
1140 PRINT AT 3,5;FIRST WN;AT 3,
7;"---";AT 2,8;FIRST N;AT 4,8;FI
RST D;AT 3,12;"▩"
1160 PRINT AT 3,15;SECOND WN;AT
3,17;"---";AT 2,18;SECOND N;AT 4
,18;SECOND D
1180 PRINT AT 7,0;"-------------
--------------------"
1200 PRINT AT 10,5;FIRST WN;AT 1
0,7;"---";AT 9,8;FIRST N;AT 11,8
;FIRST D;AT 10,12;"▩"
1220 PRINT AT 10,15;SECOND WN;AT
 10,17;"---";AT 9,18;SECOND N;AT
 11,18;SECOND D
1240 REM MIXED NUM TO IMPROPER
1260 PRINT AT 15,0;"CHANGE THE M
IXED NUMERAL▒TO AN  IMPROPER FRA
CTION"
1280 GOSUB 3980
1300 PRINT AT 15,0;"REWRITE THE
SAME DENOMINATOR"
1320 GOSUB 3980
1340 PRINT AT 10,5;"  ";AT 10,7;
"---";AT 9,8;"  ";AT 11,8;FIRST
D;AT 10,12;"▩"
1360 PRINT AT 10,15;"  ";AT 10,1
7;"---";AT 9,18;"  ";AT 11,18;SE
COND D
1380 PRINT AT 15,0;"MULTIPLY THE
 WHOLE NUMBER TIMES THE DENOMINA
TOR"
1400 GOSUB 3980
1420 PRINT AT 15,0;"MULTIPLY ";F
IRST WN;" TIMES ";FIRST D
1440 GOSUB 3980
1460 PRINT AT 15,0;"YOUR PRODUCT
 IS ";FIRST WN*FIRST D
1480 GOSUB 3980
1500 PRINT AT 13,0;"PRODUCT IS "
;FIRST WN*FIRST D;AT 14,0;"NOW A
DD ";FIRST N;" TO PRODUCT"
1520 GOSUB 3980
1540 PRINT AT 13,0;
1560 FOR A=1 TO 7
1580 PRINT "
                "
1600 NEXT A
1620 PRINT AT 15,0;"ADD ";FIRST
WN*FIRST D;" PLUS ";FIRST N
1640 GOSUB 3980
1660 PRINT AT 15,0;"YOUR SUM IS
";FIRST WN*FIRST D+FIRST N
1680 GOSUB 3980
1700 PRINT AT 15,0;"REWRITE ";FI
RST WN*FIRST D+FIRST N;" AS YOUR
 NEW NUMERATOR"
1720 PRINT AT 9,8;FIRST WN*FIRST
 D+FIRST N
1740 GOSUB 3980
1760 PRINT AT 15,0;"MULTIPLY THE
 WHOLE NUMBER TIMES THE DENOMINA
TOR OF FRACTION:2"
1780 GOSUB 3980
1800 PRINT AT 15,0;"MULTIPLY ";S
ECOND WN;" TIMES ";SECOND D
1820 GOSUB 3980
1840 PRINT AT 15,0;"YOUR PRODUCT
 IS ";SECOND WN*SECOND D
1860 GOSUB 3980
1880 PRINT AT 13,0;"YOUR PRODUCT
 IS ";SECOND WN*SECOND D;AT 14,0
;"NOW ADD ";SECOND N;" TO PRODUC
T"
1900 GOSUB 3980
1920 PRINT AT 13,0;
1940 FOR S=1 TO 7
1960 PRINT "
                "
1980 NEXT S
2000 PRINT AT 15,0;"ADD ";SECOND
 WN*SECOND D;" PLUS ";SECOND N
2020 GOSUB 3980
2040 PRINT AT 15,0;"YOUR SUM IS
";SECOND WN*SECOND D+SECOND N
2060 GOSUB 3980
2080 PRINT AT 15,0;"REWRITE ";SE
COND WN*SECOND D+SECOND N;" AS Y
OUR NEW NUMERATOR"
2100 PRINT AT 9,18;SECOND WN*SEC
OND D+SECOND N
2120 GOSUB 3980
2140 PRINT AT 10,21;"= ----"
2160 PRINT AT 15,0;"MULTIPLY ";F
IRST WN*FIRST D+FIRST N;" TIMES
";SECOND WN*SECOND D+SECOND N
2180 GOSUB 3980
2200 LET A=FIRST WN*FIRST D+FIRS
T N
2220 LET B=SECOND WN*SECOND D+SE
COND N
2240 PRINT AT 15,0;"YOUR PRODUCT
 IS ";A*B
2260 GOSUB 3980
2280 PRINT AT 15,0;"WRITE ";A*B;
" AS YOUR NUMERATOR"
2300 PRINT AT 9,24;A*B
2320 GOSUB 3980
2340 LET C=FIRST D*SECOND D
2360 PRINT AT 15,0;"MULTIPLY ";F
IRST D;" TIMES ";SECOND D
2380 GOSUB 3980
2400 PRINT AT 15,0;"YOUR PRODUCT
 IS ";C
2420 GOSUB 3980
2440 PRINT AT 15,0;"WRITE ";C;"
AS YOUR DENOMINATOR"
2460 PRINT AT 11,24;C
2480 GOSUB 3980
```

```
2500 LET DENOMINATOR=C
2520 LET NUMERATOR=A*B
2540 REM LOWEST TERMS
2560 PRINT AT 15,0;"NOW YOU HAVE
 TO PUT THE FRACTIONIN LOWEST TE
RMS."
2580 GOSUB 3980
2600 CLS
2620 PRINT AT 9,1;NUMERATOR
2640 PRINT AT 10,0;"------"
2660 PRINT AT 11,1;DENOMINATOR
2680 PRINT AT 10,6;"=    ----"
2700 PRINT AT 15,0;"COMPUTING LO
WEST TERMS...WAIT."
2720 FOR O=20 TO 1 STEP -1
2740 IF O<10 THEN PRINT AT 18,18
;" "
2760 PRINT AT 17,0;"I AM DIVIDIN
G NUMERATOR AND THE DENOMINATOR
BY: ";O
2780 LET I=NUMERATOR/O
2800 LET J=DENOMINATOR/O
2820 IF INT I=I AND INT J=J THEN
 GOTO 2860
2840 NEXT O
2860 LET NUMERATOR=I
2880 LET DENOMINATOR=J
2900 PRINT AT 9,10;NUMERATOR;AT
11,10;DENOMINATOR
2920 GOSUB 3980
2940 REM IMPROPER TO MIXED
2960 LET WHOLE NUMBER=NUMERATOR/
DENOMINATOR
2980 PRINT AT 15,0;"DIVIDE THE N
UMERATOR ";NUMERATOR
3000 PRINT AT 16,0;"BY THE DENOM
INATOR ";DENOMINATOR
3020 GOSUB 3980
3040 LET WHOLE NUMBER=INT WHOLE
NUMBER
3060 LET NEW NUMERATOR=(NUMERATO
R/DENOMINATOR)-INT (NUMERATOR/DE
NOMINATOR)
3080 LET NEW NUMERATOR=NEW NUMER
ATOR*DENOMINATOR
3100 LET NEW NUMERATOR=INT (NEW
NUMERATOR+.5)
3120 CLS
3140 REM LAST SCREEN
3160 PRINT AT 9,1;NUMERATOR
3180 PRINT AT 10,0;"------"
3200 PRINT AT 11,1;DENOMINATOR
3220 PRINT AT 10,6;"=    ----"
3240 PRINT AT 10,10;WHOLE NUMBER
3260 PRINT AT 9,14;NEW NUMERATOR
3280 PRINT AT 11,14;DENOMINATOR
3300 GOSUB 4160
3320 CLS
3340 PRINT AT 5,0;"YOU HAVE COMP
LETED YOUR PROBLEM"
3360 PRINT AT 9,0;"PLEASE GIVE S
OMEONE ELSE A TURN"
3380 PRINT AT 13,0;"ENTER Y TO R
UN THE PROGRAM AGAIN"
3400 INPUT A$
3420 IF A$="Y" THEN GOTO 40
3440 IF A$<>"Y" THEN GOTO 3460
3460 STOP
3480 REM DIRECTIONS
3500 PRINT "DIRECTIONS:"
3520 PRINT
3540 PRINT "FIRST ENTER THE NUMB
ERS ASKED.  DO NOT ENTER ZERO."
3560 PRINT
3580 PRINT "SECOND READ THE STEP
S. EVERY    STEP IS EXPLAINED."
3600 PRINT
3620 PRINT "NEXT FOLLOW THE DIRE
CTIONS ON   THE BOTTOM OF THE SC
REEN."
3640 PRINT
3660 PRINT "AT THE END I WILL PR
INT THE     ANSWER."
3680 PRINT
3700 PRINT
3720 PRINT "IT IS TIME TO START.
.."
3740 PRINT AT 21,0;"PRESS A KEY
TO START..."
3760 PAUSE 6000
3780 CLS
3800 RETURN
3820 STOP
3840 PRINT AT 10,0;"NUMBERS BETW
EEN 1 AND 99 ONLY."
3860 PAUSE 300
3880 CLS
3900 PRINT AT 21,0;"
                    "
3920 RETURN
3940 STOP
3960 REM ERASE SCREEN + PAUSE
3980 PRINT AT 21,0;"PRESS ANY KE
Y TO CONTINUE..."
4000 PAUSE P
4020 PRINT AT 15,0;
4040 FOR X=1 TO 7
4060 PRINT "
                    "
4080 NEXT X
4100 RETURN
4120 STOP
4140 REM LAST SCREEN
4160 PRINT AT 1,1;FIRST WN;" --+
"
4180 PRINT AT 0,4;FIRST N
4200 PRINT AT 2,4;FIRST D
4220 PRINT AT 1,7;"X"
4240 PRINT AT 1,9;SECOND WN;" --
-"
4260 PRINT AT 0,12;SECOND N
4280 PRINT AT 2,12;SECOND D
4300 PRINT AT 1,18;"="
4320 PRINT AT 1,25;"----"
4340 PRINT AT 1,22;WHOLE NUMBER
4360 PRINT AT 0,26;NEW NUMERATOR
4380 PRINT AT 2,26;DENOMINATOR
4400 PRINT AT 21,0;"PRESS ANY KE
Y TO CONTINUE..."
4420 PAUSE 6000
4440 RETURN
4460 SAVE "FRACTIONS"
4470 RUN
```

LISTING FOR "FRACTIONS"

$$2\ \frac{1}{2} \times 2\ \frac{1}{2}$$

$$2\ \frac{1}{2} \times 2\ \frac{1}{2}$$

CHANGE THE MIXED NUMERAL TO AN IMPROPER FRACTION

PRESS ANY KEY TO CONTINUE...

$$\frac{5}{2} \times \frac{5}{2}$$

REWRITE 5 AS YOUR NEW NUMERATOR

PRESS ANY KEY TO CONTINUE...

TYD ☆ BYTS

SPARE THAT 2068 SCREEN

Every schoolboy knows that you shouldn't leave your monitor or TV screen standing idle for a long time with a static pattern on it. (That's why the TV people caution you not to leave your MUTE command in effect for a long time--turn the volume down instead.) Anyone who ignores this warning is in danger of suffering phosphor burn-out--the part of the monitor (or TV) screen that is illuminated by the electron beam (the bright areas, that is) wears out faster than the rest of the screen, and the screen sooner or later takes on a permanent impression of the pattern.

It's sort of (but not exactly) like the after-image that you see when you close your eyes after looking at a brightly illuminated scene. Luckily, in the case of your eyes, the fatigue wears off after a while, unless the light has been bright enough to cause permanent damage.

So we are always warned to turn off the monitor when we are not using the computer. Or at least to turn down the brightness. Well, here is a mini-program that has the same effect, with only a few keystrokes required to activate it.

Include in your TS2068 program the following line:

```
9999 BORDER 0: PAPER 0: INK 0:
 CLS: PAUSE 0: BORDER 7: PAPER
 7: GO TO 1
```

Of course, you can have the PAPER, BORDER, and INK colors be whatever you like after the PAUSE. Similarly, you can locate this routine at any line number that's convenient (although you may want to precede it with a STOP, in order to keep it from being activated at the wrong time). And the final GO TO 1 can be replaced with anything you want.

So whenever you want to leave your computer for a while, you can put the monitor to "sleep" by GO TO 9999, and everything will go dark until you touch a key. When you do touch any key, the program restores the colors you want, and resumes operation at the line you've chosen.

You can make this even easier by including in Line 1 (or some other line) the command

```
LET G=9999
```

Then your "hibernate" command is just a simple GO TO G, which is about the easiest command there is. If you do decide to do things this way, be sure that you include a provision to redefine G=9999 whenever the program RUNs or CLEARs. Or you can include SLEEP as a menu option.

You can do something similar with the TS1000, by having it PRINT a screenful of inverse spaces, but I don't know of any way to protect the BORDER area.

LARKEN 3 1/2" DISK SYSTEM

for $199.95

Package includes ...
- Larken 2068/Spectrum Disk Interface and LKdos Cartridge
- 1 - 400K 3.5" Mitsubishi Disk Drive and Power supply
- or 1 - Single sided 5 1/4" Disk drive and Power supply
- 1 Disk of Disk converted programs to start you off.
- Other Drive combinations available .
- Ready to Run Floppy disk System . 90 day guarantee
- Limited Time Offer, Until 3.5" 's are gone .

(prices are US $ add $10 S&H)
LARKEN ELECTRONICS, RR#2 NAVAN ONTARIO, K4B-1H9 CANADA
(613)-835-2680

AFR SOFTWARE (R)

Presents:

Powerful And Inexpensive
Business Software
For "Timex-Sinclair"
Computers

WORD PROCESSING
T/S-TEXT 2000 ..... $19.95
ZX-TEXT $19.95

SPREADSHEET CALCULATOR
T/S-CALC 2000 ..... $19.95
ZX-CALC $19.95

CYCLE ACCOUNTING
T/S-ZX Financial Report Generator $29.95
Printout Of Same $13.00

APPOINTMENT SCHEDULER
T/S-CALENDAR 2000 ..... $19.95
ZX-CALENDAR $19.95

Send S.A.S.E. For Free Catalog
Or Check Or Money Order To:
A.F.R. SOFTWARE
1605 Pennsylvania Ave.
No. 204
Miami Beach, FL 33139
(305) 531-6464
"FLORIDIANS ADD SALES TAX"
Dealer Inquires Invited

TYD☆BYTS

PUT THAT TAPE RECORDER TO WORK

If you use a portable (battery-powered) tape recorder for your SAVEs, or if you have a tape deck in your car, you may be under-using a valuable resource.

ISTUG (the Indiana Sinclair Timex Users' Group) recently started holding its meetings in a branch library in Indianapolis. (Hint to other users' groups: the local public library is often very generous with the use of its space by hobby groups, as long as you have a large enough attendance.)

Well, I don't know Indy very well, and my memory isn't what it used to be (in fact, I don't remember whether it ever was). So I could visualize all sorts of traffic difficulties while I tried to cruise up I-465 with one eye on the map that Frank Davis had published for us.

What to do? I read into my cassette recorder a detailed description of the instructions, and then played them back as I drove along.

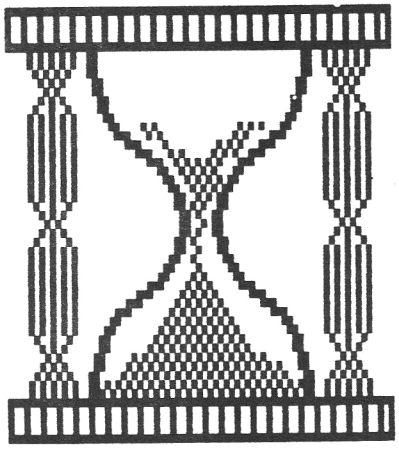
Other variations of this technique might be effective, like recording instructions from the telephone. Or, if it doesn't give offense or stage fright, you can use your tape recorder with the local resident that you stop on the street to ask for instructions. Even if he or she does object to being recorded directly, you can always repeat the instructions into the recorder, phrase be phrase, as you receive them.

But in any case, politely ask your informant's permission first, or you might be taken for a CIA agent.

TIME IS RUNNING OUT!

Get your Back Issues of Quantum Levels and SyncWare News today while the supply lasts!
ONLY $2 each! Save even more...Buy any 6 for $10!
Some issues available in very limited quantities.

CK, MO, VISA/MC Accepted



SEND ORDERS TO: BACK ISSUE DEAL
The SyncWare Group
602 S. Mill St.
Louisville, OH 44641

Include your Name, address, telephone number, and a list of the issues desired.

A few of the difficulties found are listed here. First, SpeedScreen occasionally conflicted with Solution, either fouling up the screen or locking the keyboard (??). The documentation instead mentions the use of Digital's Lightning as completely compatible. The problems were rare and I assume easily fixable.

Multitasking worked well except for a few glitches. After entering a Basic command, the cursor always switched back to DOS (Solution). This inadvertent task switch also occurred after some command type entries in other multi-tasked programs.

Digital made provisions for their QL toolkit print screen option to also work in DOS, but did not allow for other toolkit ALT key options to be available (such as the QJump Toolkit II screen print command). Also, the DOS print command sent only 'x's to an EPSON MX80 printer.

The XOVER program definitely did not like Speed Screen as it would lock the keyboard during a directory printout if the end of screen was reached (it worked fine without SpeedScreen. Error messages were also lacking during format problems in this utility (unlike the quality error handling of The Solution).

SUMMARY

As far as emulators go, Solution is a definite winner if you need such a program. You must get used to the display scanning method and the delays in waiting for your keyboard entries to appear.

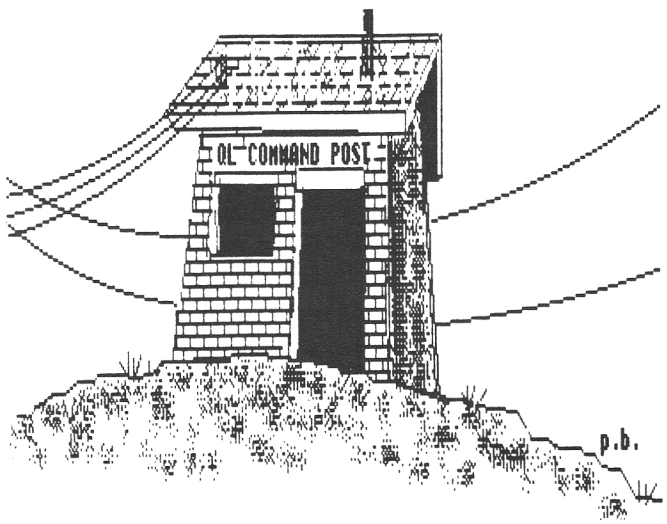
Plans to be working on something else might be a good idea while running some programs. Multitasking is not practical unless there is absolutely no hurry for the DOS program to finish).

The problems are few and easy to avoid. In perspective, Solution has an amazingly low level of bugs when viewing both the complexity of emulators and the infamous number of quirks in DOS. Solution actually works around some of these quirks.

It must also be understood that purchase of The Solution does not include lessons in DOS. Digital Precision, in their documentation, makes no little effort to help you understand this (justifiably so). Additionally, if a DOS program does not run properly you must first confirm that it runs properly on other IBM clones before bringing it to the attention of Digital as a problem of theirs (it is most likely a problem with your DOS program).

The Solution is easy to use and, with an understanding of its inborn speed limits and the possibility of 'illegal' DOS programs, can be a useful tool if you need IBM capability. Even at $150 plus the DOS program cost, it is a bargain in comparison to a $650 to $900 XT black and white system. Not to mention who wants a bulky and run of the mill IBM clone taking up table space that could be better put to use by your QL system.

And, oh yes, the use of DOS with Solution is an excellent reminder of the versatility and advanced concepts of the QL operating system and software.



PROGRAMMING BY PROCEDURE

Peter Hale

Part IV: Decision Making

In ARCHIVE the only obvious decision making command is: IF ... ELSE ... ENDIF. It works as you would expect it to work in SuperBASIC, but ENDIF is always required. It must also do the work of SELect. There is no THEN. Instead use a colon.

However, the WHILE...ENDWHILE loop command is also a versatile decision-maker dispensing with FOR...END FOR and REPeat...END REPeat.

The condition to exit a WHILE ... ENDWHILE loop is set at the beginning of the loop.

For example:

```
while not eof() : rem eof()=end of file
while x$<>""    : rem exit if x$=""
while a$='m'    : rem exit if a$<>"m"
while n<7 or m<5
while x>=y*z and x<=(y+1)*z and not eof()
```

The loop will continue until the specified condition is reached.

while 1 starts a loop that will never end, but it is useful for handling errors.

while 0 is for a loop that will never start or conversely will always end.

However, the following pair of procedures have wide application for error handling:

```
proc inkey
  let n=1
  while n
    error in_key
    let n=errnum()
    if n: print "Error number ";n: endif
    endwhile
  endproc

proc in_key
  print at 2,2;'Enter letter: ";
  let z$=getkey(): print z$;
  endproc
```

From the keyboard type 'inkey <ENTER>'.

On entering the loop 'n' is always 1; the procedure in_key is called on first entering the loop, but the command ERROR that preceeds it prevents the program from crashing in the event of an error. Instead, it returns to the procedure inkey and continues the loop.

If a printing character or ENTER is pressed in response to the prompt, the errnum() value 'n' resets to zero and the loop is exited.

In SuperBASIC the comparable WHEN ERROR keyword is not documented because it does not reset to zero in the JSU rom. (If you have Toolkit II, the bug is fixed.) In ARCHIVE the error trapping works.

However, any error (i.e., ESC or a function key) will generate an error that (1) returns you to inkey (with errnum()=92) and loops to in_key and (2) never lets you out of the loop until you make an acceptable keystroke.

Two absolutes about exiting loops:

1) Although an exit condition has been reached, a loop will continue up to ENDWHILE before exiting. So put the instruction that changes the variable effecting exiting where it should be. See the difference:

   ```
   let x=0:while x<20:print x:let
   x=x+1:endwhile
   let x=0:while x<20:let x=x+1:print
   x:endwhile
   ```

2) while not eof() is a common way of limiting a loop when you wish to perform an operation on all records in a file. Such a loop-limit will exit at the end of the file.

However, it will never exit unless the instruction NEXT is within the loop. Without NEXT the loop will continue to process the same record forever. Put NEXT just prior to ENDWHILE. When the final record is called, the loop will not exit until the final record itself faces the NEXT command.

LOOPS WITHIN LOOPS

In SuperBASIC you may nest loops and so you can in ARCHIVE. Consider however, that when operating on an entire file (i.e. while not eof()) a nested loop must also include the

'not eof()' as in:

```
while not eof()
  while x=2 and not eof()
  endwhile
endwhile
```

Otherwise the last record will be operated on ad infinitum, until the cows come home or until there is a power failure.

Tips for datanoids:

ARCHIVE has no function for generating random numbers. Why would it? The essence of data storage is exactness. However, a quasi-random number can be used to generate non-data values, such as ink or paper colors.

Try this procedure:

```
proc rnd
 let n=val(time()(8)): rem (8) is the second
 if n<2: let n=n+3: endif : rem avoids black
 if n>7: let n=n-3: endif : rem within color
 endproc
```

Rewrite the in_key procedure above:

```
proc in_key
  rnd: print at n,2;ink n;'Enter letter: ";
  let z$=getkey(): print z$;
  endproc
```

Homework: Rewrite a SuperBASIC program to operate in ARCHIVE. It is not necessary to have a data file; indeed it's better to do a program with no stored data. Avoid graphics as ARCHIVE has limited resources

PLAYING WITH ELECTRICITY

Harvey Taylor

When I was in Seattle at the Sinclair Fest, I discovered people had questions about the memory layout of the QL. In particular, the command RESPR was not understood.

The QL memory map has two aspects, what the hardware allows & what QDOS expects. The hardware aspect is straightforward; the 68008 has 20 address lines which define a space of 1 million locations. These are arranged as shown in Table_0.

The memory map as seen by QDOS is independent of the available ram, in relative terms. There are several major sections. These are as shown in Table_1.

When you first power up the Resident Procedure area and the Transient Program area are empty. As you extend SuperBasic or run jobs these System Variables are changed accordingly. For example, a common operation in a boot program will be something like;

```
100 ADDR = RESPR(1024)
110 Lbytes mdv1_EXTCODE,ADDR
120 CALL ADDR
```

The instruction RESPR(1024) has the effect of lowering SV_RESPR 1024 bytes, if there

TABLE 0

K	DEC	HEX	USE
1024K	1,048,576	00100000	Top of Address
		000FFFFF	Space +1
960K	983,040	000F0000	
896K	917,504	000E0000	16 - 16K Peripheral
832K	851,968	000D0000	Cards
768K	786,432	000C0000	
		000BFFFF	
704K	720,896	000B0000	
640K	655,360	000A0000	
576K	589,824	00090000	
512K	524,288	00080000	512K Expansion Ram
448K	458,752	00070000	
384K	393,216	00060000	
320K	327,680	00050000	
256K	262,144	00040000	
		0003FFFF	
192K	196,608	00030000	
			128K On Board Ram
128K	131,072	00020000	
			I/O Hardware
64K	65,536	00010000	
			ROM Expansion
48K	49,152	0000BFFF	
			ROM
0K	0	00000000	

is memory to spare. This is not unlike the old ZX81 technique of lowering ramtop.

It is easy enough to write a PROCedure which reads the System Variables and tells you how the memory allocations of your QL are changing as you use it. The PROCedure TELL below does this if you don't want to write your own.

Table 1

System Variable	Function	
SV_RAMT	Top of Ram	
	Resident Procedures	
SV_RESPR		
	Transient Programs	
SV_TRNSP		
	SuperBasic Data & Programs	
SV_BASIC		
	Slave Blocks use this Free memory	
SV_FREE		
	Channel Definition Blocks Device Drivers & Program use of Common Heap.	
SV_HEAP		
	System Variables	
SV_BASE		$28000
	Default Video Ram	
RAM_BASE		$20000

```
100 REMark Print Sys_Var & Mem_Alloc
110 :
120 DEFine PROCedure PRT_STATS
130   VR_BASE=131072
140   SV_BASE=163840
150   SV_END =164992
160   SV_CHEAP=SV_BASE+4
170   SV_FREE =SV_BASE+12
180   SV_BASIC=SV_BASE+16
190   SV_TRNSP=SV_BASE+20
200   SV_RESPR=SV_BASE+28
210   SV_RAMTP=SV_BASE+32
220   a=PEEK_L(SV_CHEAP)
230   c=PEEK_L(SV_FREE)
240   d=PEEK_L(SV_BASIC)
250   e=PEEK_L(SV_TRNSP)
260   f=PEEK_L(SV_RESPR)
270   g=PEEK_L(SV_RAMTP)
280   PRINT 'System Variables'
290   PRINT 'SV_RAMTP =';g
300   PRINT 'SV_RESPR =';f
310   PRINT 'SV_TRNSP =';e
320   PRINT 'SV_BASIC =';d
330   PRINT 'SV_FREE  =';c
340   PRINT 'SV_CHEAP =';a
350   PRINT 'SV_BASE  =';SV_BASE
360   PRINT 'VR_BASE  =';VR_BASE
370 END DEFine PRT_STATS
380 :
390 DEFine PROCedure TELL
400   PRT_STATS
410   PRINT¢'MAJOR SYSTEM MEMORY ALLOCATIONS'
420   PRINT 'TOTAL RAM AVAILABLE: ';g-VR_BASE
430   PRINT 'RESIDENT PROCEDURE AREA: ';g-f
440   PRINT ' TRANSIENT PROGRAM AREA: ';f-e
450   PRINT '        SUPERBASIC AREA: ';e-d
460   PRINT '       FREE MEMORY AREA: ';d-c
470   PRINT '       COMMON HEAP AREA:
';c-SV_END
480   PRINT '       SYSTEM VARIABLES:
';SV_END-SV_BASE
490   PRINT '              VIDEO RAM:
';SV_BASE-VR_BASE
500 END DEFine TELL
```

oooooooooooooooooooooooooo

QUANTA

QL Users and Tinkerers Association
Independent QL/Thor Users Group

Worldwide Membership is by subscription only and offers the following benefits:
Monthly Newsletter (up to 40 pages)
Massive Software Library - Mostly FREE
FREE Helpline- Workshops - Bulletin Board
Regional Sub-Groups - one near you?
Advice on Hardware & Software problems
Discounts from most major suppliers
Subscription just £14 for UK members
Overseas Subscription £17
Barclaycard - Visa - Access - Mastercard

Further details from the Hon. Secretary:

PHIL BORMAN,
15 GROSVENOR CRESCENT,
SOUTH HUMBERSIDE,
DN32 0QJ
Tel: (0472) 49850

TRA_BAS 1.1:RS-232 Translation

Al Feng

RS-232 TRAnslations may seem a bit esoteric to some users, but the implementation of an RS-232 TRAnslation can resolve minor printer interface problems. Generally, your QL (and most other personal computers) telecommunicates with other computers along with communicating with other devices using the non-standard (!) ASCII code. Fortunately, the significant alphanumeric series is standardized. Nonetheless, you may find it expedient to execute one, or many, pairs of serial translations.

A simple example would be if you were more comfortable with a German keyboard layout which transposes the 'Y' and 'Z' characters. The install_bas program used to set up the printer_dat file for use with the Quill program can be used for this TRAnslation procedure. But, if you are like me, the data listings at the bottom of the screen's page is cryptic; and the USER'S GUIDE is not clear about how to employ this feature. Additionally, there may certainly be occasions when you will wish to implement a TRAnslation exclusive of one of the Psion programs.

Further, while many printers contain several foreign language character sets, the accesss to the special characters may be on unfamiliar, "non-standard" keyboard locations. If you wish to use your QL, or other computer, for foreign language word processing, and the computer does not have the appropriate firmware, the irregular placement of the special characters may make the process tedious. Fortunately, Jan Bredenbeek (after Kees van der Wal) has provided QL users a fairly easy way to implement RS-232 TRAnslations.

TRA_BAS 1.1 is an enhancement of Jan Bredenbeek's TRA1_bas program (October 1986 QUANTA). Where the onscreen prompts of TRA1_bas tend to be opaque, I have tried to make them easier for you to follow. Also, I have changed the "tra" extension to a suffix to facilitate file recognition, and have added the facility to name the file on an ad hoc basis.

To implement a TRAnslation of the 'Y' and 'Z' characters you would want to tell your printer to interpret ASCII decimal code 89 as 90, and 90 as 89. Also, you would want to tell your printer to TRAnslate ASCII 121 for 122, and vice versa, to represent the lower case letters 'y' and 'z', respectively. Other character coding would be dependent on your particular printer's code position. You can find a listing of the QL's ASCII character set and corresponding keys in pages 5-9 of the "Concepts" section of your QL USER GUIDE.

The program's on screen CODE requested is the "old" decimal code. The REPLACE_CODE is the "new" code that you want sent via SER1 (or, whatever device you have specified). The "number of replace codes (1-3)" allows the user to send multiple CHR$ in place of the original implementation. The total TRAnslations allowable is determined by the DIM statement and 'loop1.'

The information is reloaded/used by including the following lines in your new boot and/or program:

```
n12  a=RESPR(###)
n14  LBYTES mdv1_codename_tra, a
n16  TRA a
```

You will have to write this information down on a piece of paper (i.e., the '###' and codename you have chosen) if you want to use it in a program.

If you have Lib_29 (Howard Clase's excellent FOUNT_bas program) you can modify your Quill/QLWP program for use as a foreign language word processor to be used with the IBM WHEELWRITER 5 (or, some other printers if the desired foreign language printwheel is available) and its various foreign language printwheels for letter quality output. You will find it convenient to use the TRA_bas program to "correct" the serial output for the "[ ] and shifted [ ]" keys.

If you happen to have IBM's Russian Cyrillic printwheel, you will notice that it has a "flaw" as one cannot get capital letters for either the Cyrillic 'B' or 'yU' when the WHEELWRITER 5 is used as a printer in-

stead of a typewriter. So, I suggest using the TRA_bas program to implement lower case variations instead of the "nonsensical" superscripted '2' and '3' which will appear; and, to TRAnslate the 'X' and 'hard sign'.

(NB: I have called IBM DIRECT, and they had absolutely no knowledge what I was talking about with regard to the printwheel. I have tried just about every "language code" that is available, but have not been able to get all the characters to print when using the Wheelwriter as a printer. You may encounter a similar problem with some of the other printwheels.)

[A 96 character printwheel can only access 94 characters, which is the standard ASCII character set. The other two are swapped by changing hard switches on the printer itself.--ed.]

Non-typist may choose to use the two programs (TRA_bas 1.1 and Fount_bas) in tandem to implement a DVORAK keyboard. The QWERTY keytops can be easily swapped to the appropriate location by simply prying the keytops off with a butter knife.

A more exotic use of this TRAnslate utility would be to turn your QL into a "Captain Midnight decoder ring" to scramble or otherwise decode the data being sent either over telephone lines or on magnetic data files. Of course, the receiver and sender of this scrambled data would have to have a copy of the appropriate SuperBASIC data, and the same codename tra file.

If you do not use floppy disks, replace 'flp_' in the LISTing below with 'mdv_.'

```
100 REMark ***************************
110 REMark *           TRA_BAS 1.1     *
120 REMark *         TRA-command for   *
125 REMark *       RS-232 translations *
130 REMark *Jan Bredenbeek, Hilversum *
135 REMark *       (1986 April 13)     *
140 REMark *    amended by Al Feng     *
145 REMark *  PLATYPUS Software 1988   *
150 REMark ***************************
160 OPEN#5,SCR_512X255a0X0:PAPER#5,7:CLS#5
170 OPEN#6,SCR_460X14a30X7:PAPER#6,7:INK#6,4:
    BORDER#6,1,7
180 MODE 4: WINDOW 450,216,30,20:PAPER 3:INK
    6
200 DIM v(3)
210 start=RESPR(1024)
220 POKE_W start,19195:  REMark $4AFB
230 table_a=start+6: table_b=start+262
240 POKE_W start+2, table_a-start: POKE_W
    start+4, table_b-start
250 FOR i=0 TO 255: POKE table_a+i,i
260 pointer=table_b+i
270 ncodes=0 : CLS
290 PRINT#6¢¢," RS-232 translations
    (TRA-command)      by Jan Bredenbeek"
300 PRINT¢, "  <27> to STOP & SAVE code when
    finished with ALL changes"
310 REPeat loop
320    INPUT¢,,"Code to be changed: "!a$
330    IF a$=27 THEN GO TO 540
340    IF a$="" THEN EXIT loop
350    REPeat loop1
360 INPUT "   Number of replace codes (1-3)?:
    "!n
370 IF n>0 AND n<4 THEN EXIT loop1
380    END REPeat loop1
390    FOR i=1 TO n
400    INPUT¢,,"   REPLACE_CODE"!(i);"?:
    "!v(i)
410    END FOR i
420    IF n=1 THEN
430      POKE table_a+a$,v(1)
440      ELSE
450      POKE table_a+a$,0: POKE pointer,a$
470      FOR i=1 TO n: POKE pointer+i, v(i)
480       ncodes =ncodes+1:pointer=pointer+4
500   END IF
510   END REPeat loop
520     POKE table_b,ncodes
530     TRA start,0
540 CLS:INPUT¢¢¢!!!!!"Name of data code to be
    saved:"!name$
550 SBYTES "flp1_"&name$&"_tra", start,
    pointer-start
560 PRINT¢,"You may reload your TRA_codes
    with (e.g., include in BOOT):"
570 PRINT¢,,"  xx0
    a=RESPR(";pointer-start;")"
580 PRINT,, "  xx2   Lbytes
    flp1_"&name$&"_tra,a"
590 PRINT,, "  xx4   TRA a"
600 REMark end of LISTing TRA_bas 1.1
```

Many thanks to Jan Bredenbeek, Dr. Howard J. Clase, and Kees van der Wal for making foreign language word processing possible for QL users.

oooooooooooooooooooooooooooo

oooooooooooooooooooooooooooo

Think About This!

The Oklahoma Syndrome!

Did you ever wonder just how fast the speed of light (and electricity) is? Well, at 186,000 miles/ second thats lots of inches/second. In fact it is 11.78 inches /nanosecond. Now add harmonics (think of 12 individual notes in an octave on the piano keyboard). If you go 1/12 up the octave you come to the next pleasing sound. It is at this point where the sounds add up at certain times and cancel out at other times to create the (generally pleasing) sound you hear.

Now go back to the electricity. 1/12 of the speed of light/nanosecond is .982 inches. If a logic transition takes 10 nanoseconds, you may start having trouble if your trace length is over 10 inches. Well guess what? Most fast computers suffer from this phenomenon (yes that includes PC clones too). Although there are ways to terminate bus lines and impedance match etc., it is truly a designers nightmare.

The Punch Line

If Sinclair had made the QL square (like most other high speed computers) instead of rectangular with a pan handle, he may well have avoided going to an issue 7 PC board way back when. (The clock would have worked too!)

UPDATE!

Taken to the Limit

A Quarterly Magazine for the Users

Supporting the Sinclair QL, Z88, and TS-2068
Subscription $15.00 Year. UPDATE Magazine,
1317 Stratford Ave., Panama City, FL 32404

Software for the QL that really works

MAILBAG An address database you won't believe. Labels, Rotary files, Zip sort, tickler file, many **auto features**, & more...256K

TAX-I-QL Spreadsheet for **IRS** tax returns.....384K

QLANDLORD Manage 99 buildings or units. Do Sched E..256K

$19.95 ea pp on 5 1/4" DSQD
(+$2.00 on microcartridge)

AND The most comprehensive list of independent North American software for the QL.

Agents for ZX/TS **SiriusWare**
4K Wordprocessor with
TS 2040 lower case printer driver

Orders, Catalogues & Dealer info

EMSoft
P.O. Box 8763, Boston, MA 02114
(617) 889-0830
VISA/MC ok

TRY THIS!!

As a side note, those of you who have a QL schematic please follow the data bus path around the layout. Do you notice how the ZX8301, ram chips and ZX8302 are connected toget- her and the CPU, ROMs and external bus are connected together?

Take an Ohm meter and touch between any data line on the 8301 and 8302. Interesting? Now try the CPU and 8302 (should set off a light bulb upstairs).

Because of early problems, Sinclair moved the 8302 from the buffered ram area to the unbuffered CPU area (good move, but no cigar). Now why do you suppose the Clock quit working as designed? Although it improved the integrity of the 8301, it worsened the external data bus significantly.

Some QL's have trouble when a simple little eprom board is plugged in, let alone an extra foot of ram and disk interface. Fortunately, most interfaces buffer the data bus.

To MG or not To MG

Many questions have come up concerning the value in upgrading the American JSU Roms to the British JS or MG Roms. Sharp's reports selling Eprom replacement boards with mostly JSU Eproms on them. I suspect this is due to the "Leave well enough alone" syndrome. So, how do you determine whether it is OK to upgrade?

The answer is straightforward. If you use an RGB monitor, you can use the MG Roms. In fact, the only ones who cannot easily use MG Roms are those who only use a TV on their QL. (This is a bug in the JSU that you have unknowingly worked around. All of the other ROM versions are screen compatible with each other.)

Why bother? The MG Rom has the latest versions of QDOS and SuperBasic and a few new features have been added. Since the QL multitasks, the power to move jobs and consolidate memory to avoid the dreaded "Out of Memory" error has been added. You instead get "I've forgotten what I was doing" error. (Humorous eh?) This means that jobs have been moved and pointers reset. In the JSU you would most likely have to reset and reboot at this point. There are some other less significant changes to SuperBasic that make it even more unlike the QL Manual. We are compiling a list and will print it soon.

If you have found discrepancies in the manual, drop us a line and we'll print them. In the mean time consider MGing at the first possible chance. The latest ROM is the MGSJ. This supercedes the MGUK. The SJ doesn't stand for stupid jerk or even JS backwards, but rather for Stephen Jonke (an American no less) the young QLer that incorporated the latest fixes.

Quantum Levels

602 S. Mill St., Louisville, OH 44641

Bulk Rate
U.S. POSTAGE
PAID
Louisville, OH
Permit No. 40

ADDRESS CORRECTION REQUESTED

DATED MATERIAL
DO NOT DELAY

TO SUBSCRIBE to Quantum Levels, send $18.95 for 6 issues to the address printed above. All correspondence of an editorial nature should be sent to:

Quantum Levels, Editorial Dept.
9016 Flicker Place
Columbia, MD 21045

TIPS AND TECHNIQUES FOR QL COMPUTERISTS

ALL CONTENTS COPYRIGHT 1989, The SyncWare Group